MultiModal Dataset for Machine Learning Applied to Telecommunications

Arthur Nascimento, Walter Frazão, Ailton Oliveira, Diego Gomes and Aldebaro Klautau

***Abstract*— Gathering channel data to test telecommunication systems is an essential step to guarantee the quality of the product. However, it can be a slow process and demand a considerable amount of effort and investment since it is costly to make field measurements of mmWaves. Having a ready dataset at disposal make things way faster and cheaper, allowing a developer to focus on more specific tasks. This paper presents an entire multimodal dataset with different kinds of information like channel communication, urban traffic and obstacles position, got from two realistic computer simulations made in two different city models: Beijing and Rosslyn. It also includes detailed information on how each data is stored.**

***Keywords*— 5G, mobility, deep learning, ray-tracing, LIDAR.**

I. Introduction

Taking advantage of machine learning tools to improve telecommunications is one of the 5G proposals, but it demands a large amount of available data to test algorithms and evaluate the performance of systems. Besides that, millimeter waves (mmWave) measurements for researches in 5G multiple-input multiple-output (MIMO) demand costly campaigns in elaborated external areas [1].

One of the biggest challenges for 5G data generation is the fact that real data related to wave propagation is hard to acquire due time required to prepare the scenery and configuration of equipment, and cost, due to the acquisitions of specialized hardware and human resources. In this context, synthetic data is a viable alternative once its generation has a relatively low cost and scalability of environments.

The work in [2] presents a dataset with small scale parameters taken from ray-tracing (RT) simulations, in which the snapshots of the simulation are time-related, like frames in a video. The dataset presented in [3] also contains time-related data, but it uses images instead of propagation parameters.

This paper presents a novel dataset generated using multiple data sources. It starts using RT simulations in a scenery of urban mobility [2], taking advantage of the RT capacity to handle 5G requirements such as spatial consistency, which has been a difficulty in stochastic modelling. Then, it uses LIDAR scans, an obstacle detector, because of the mobility involved, to provide spatial data about the environment around the user equipment (UE) and the base station (BS). Finally, in order to use vision to assist wireless communications [3], a methodology was developed to generate footage from the data previously mentioned, using realistic textures and illumination.

Arthur Nascimento, Walter Frazão, Ailton Oliveira, Diego Gomes and Aldebaro Klautau are with LASSE - Telecommunications, Automation and Electronics Research and Development Center, Belém-PA, Brazil. E-mails: arthur.nascimento@itec.ufpa.br, walter.costa@itec.ufpa.br, ailton.pinto@itec.ufpa.br, aldebaro@ufpa.br. Thanks to CNPq and CAPES for partial financial aid.

II. Framework description

This work presents a dataset composed by LIDAR scans, images, propagation data and geographic coordinates.

The overall procedure to generate realistic data is based on the methodology in [2], which consists of the extraction of scenery based in real locations [4], and export it to a RT simulator called Wireless InSite (WI), which is used in parallel with the open-source Simulation of Urban Mobility (SUMO) [5], that is responsible for creating the mobility feature. A python orchestrator coordinates all this [2], generating the communication data with time correlation. The acquisition of LIDAR data integrates the RT simulation, returning a point cloud [6], obtained with Blender Sensor Simulation (BlenSor). Lastly, the image acquisition is made taking advantage of the LIDAR environment, with photo-realistic textures added to the 3D objects, where cameras are positioned in the base station (BS), and an image is rendered for each simulation scene.

Note that the scenery represented across the software above are “paired”, which means that objects dimensions and positions match between the systems, so the different data types represent the same situation.

III. Data Description

A. LIDAR Data

LIDAR is a type of sensor that works with light reflection, being able to measure distances untill the first obstacle. Its data can be stored as a .pcd file (point cloud) that contains the exact spots that light reflected, being able to create a 3D map of the sensor surroundings.

A traffic simulation was created on Blensor in the corresponding scenery to obtain precise LIDAR data from the scans. Blender is a known free software made to allow 3D modelling, 3D animations, texturing and rendering. Blensor is a modified version of it that allows simulation of range scanning devices.

A point cloud is obtained for each car with a LIDAR sensor attached, repeating this process for all the simulation scenes.

The LIDAR scans were performed with the following configurations:

- Model: Velodyne HDL-64E2
- Scan distance(m): 120
- Scan angle resolution: 0.17
- Viewing angle: 360

Quantization is done to convert the results into a format easier readable by a neural network, converting the pcds information into Obstacles Matrices, that represents the zone of interest of the road. Obstructed spots are represented as 1, non obstructed spots as 0, the base station as -1, and the transmitter is represented as -2.

B. Image Data

Using cameras is an option to understand the traffic situation at a specific time with a cheaper device. Its information is not very detailed like LIDAR scans but can work as support for a precision gain.

Blender has a *render* function that simulates a camera, taking pictures like would be done by a real one. In the generated scenery, for every simulation frame, three photos are taken from cameras positioned in the BS. Each photo is stored in a Portable Network Graphic (PNG) colour image.

C. Ray Tracing Data

The data which describe the propagation rays are organized in a 4D structure $N_s \times N_{tx,rx} \times N_{path} \times N_p$, which are respectively the number of scenes, number of Tx and Rx pairs, the maximum number of rays (paths) and the path parameters. Altogether, there are eight parameters available: Received power (dBm), time of arrival (seconds), angle of arrival and departure (both azimuth and elevation), Flag '1' for a line of sight ray, and '0' for non-line of sight, and path phase (degrees). The associated codes for processing this data made available at [7], including MIMO channels examples.

D. Coordinates

This data serves the geographical localization of the vehicles in the modelled scenery. This information can be interesting for applications which assume that GPS information is available. This data is provided in a comma-separated values (CSV) file, wherein each row is found: The index of the episode, the index of the scene, the Rx index, vehicle identification, vehicle type(car, bus or truck) and 'X,Y' coordinates

IV. Results

We generated multimodal data for two scenery, representing areas of the cities Rosslyn-USA, and Beijing-China, chosen because they have good maps of their buildings [4]. The parameters of the simulations are shown in Table I, where T is the sampling interval, Ns and Nepi are the number of scenes and episodes, respectively. In both scenery were considered ten mobile Rx's, and the transceivers configured to transmit/receive a carrier 60 GHz.

The Fig. 1 shows an image taken from Rosslyn scenery, already with the textures and with vehicles travelling at the lanes.

TABLE I

CONFIGURATION OF THE SCENERY REPRESENTED WITH MULTIMODAL DATA.

Local	Frequency	T	N_s	N_{epi}	Mobile
Rosslyn	60 GHz	1 ms	1	2086	True
Beijing	60 GHz	1 s	40	50	True

Fig. 1. Rosslyn scenery with textures applied on Blender.

Tests were done using a deep learning neural network focused on the selection of best beam pair between a transmitter and a receiver, in which the input data were processed LIDAR data, images and coordinates [6]. The results can be seen in Fig. 2, which shows the Top-K accuracy.

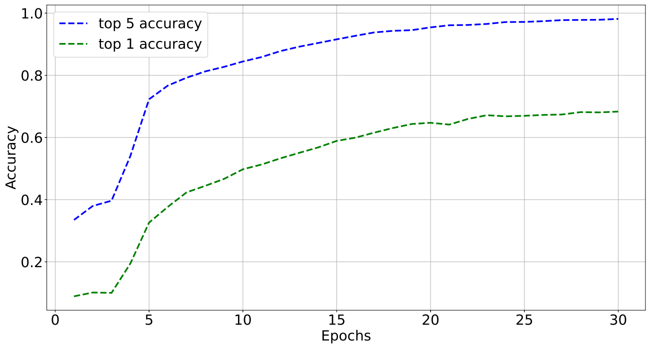

Fig. 2. Beamforming Top-K accuracy along the epochs using the RayMobTime dataset.

V. Conclusions

The dataset presented in this work is available at [8], and was made in order to help developers have easy access to different types of data obtained from an urban environment, trying to accelerate the advance of projects that would need a longer time to be finished due to the need to produce its simulations.

References

[1] G. R. MacCartney, H. Yan, S. Sun, and T. S. Rappaport, *A flexible wideband millimeter-wave channel sounder with local area and NLOS to LOS transition measurements*, May 2017.
[2] A. Klautau, P. Batista, N. Gonzalez-Prelcic, Y. Wang, and R. W. H. Jr., "5G MIMO data for machine learning: Application to beam-selection using deep learning," pp. 1–1, 2018. [Online]. Available: http://ita.ucsd.edu/workshop/18/files/paper/paper_3313.pdf
[3] M. Alrabeiah et al, "Viwi: A deep learning dataset framework for vision-aided wireless communications," 2020.
[4] A. Oliveira, M. Dias, I. Trindade, and A. Klautau, "Ray-tracing 5g channels from scenarios with mobility control of vehicles and pedestrians," 2019.
[5] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo - simulation of urban mobility: An overview," in *in SIMUL 2011, The Third International Conference on Advances in System Simulation*, 2011.
[6] M. Dias, A. Klautau, N. González-Prelcic, and R. W. Heath, "Position and lidar-aided mmwave beam selection using deep learning," 2019.
[7] "Raymobtime Github," https://github.com/lasseufpa/Raymobtime.
[8] "Raymobtime," https://www.lasse.ufpa.br/raymobtime/.